

REMARKS

Applicant respectfully requests allowance of the subject application.

Claims 1, 3, 6, 14, 26 and 32 are amended.

Claim 9 is cancelled.

Claims 1-8 and 10-36 are pending.

In view of the following remarks, Applicant respectfully requests that the rejections be withdrawn and the application be forwarded along to issuance

§ 101 Rejection

Claims 1-5 stand rejected under 35 U.S.C. § 101 as directed to nonstatutory subject matter. The Applicant respectfully traverses the rejection.

Section 2106 of the MPEP gives an example of a computer-related process limited to a practical application in the technological arts as follows:

A computer method of optimally controlling transfer, storage and retrieval of data between cache and hard disk storage devices such that the most frequently used data is readily available. *See MPEP, Section 2106.*

Claim 1 also recited a computer-related process that involves a practical application, and thus is also statutory, e.g., “emulating an operation” and “permitting the emulated operated to access a contiguous portion of emulated memory”. Claims 2-5 are also statutory based on similar reasoning. Thus, withdrawal of the rejection is respectfully requested.

§ 102(b) Rejection

Claims 1-2, 4-28, 31-34 and 36 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,073,968 to Morrison et al. (hereinafter "Morrison"). The Applicant respectfully traverses the rejection.

Morrison describes a method and apparatus for marking emulation analysis states. Additional memory for holding marking tags is used for providing additional information regarding states acquired by an emulator during tracing for dequeuing. The marking tags are determined according to a predetermined coding scheme, loaded in a marking memory, and acquired during tracing along with the fetched instruction states. The combination of addresses, data, status, and the additional marking tags is converted into a list of states which correspond to the test program executed by the target processor means. *See Abstract.*

To generate these tags, Morrison describes, in reference to FIG. 7, a flowchart of a marking software state machine module. After this module is called, the mark pointer is initialized to the beginning address for the range of addresses to be marked, or start address (block 98). The read pointer is also initialized to the start address (block 100). The opcode information is extracted from the location indicated by the read pointer (block 102) and the appropriate mark code is generated by referencing a lookup table (block 104). This lookup table contains a predetermined coding scheme for marking emulation analysis states with the following definitions:

000--null (has not been marked or has been affected)

001--low byte is the only opcode

010--high byte is the only opcode

011--both bytes are opcodes

100--neither byte is an opcode but it is marked

If the opcode is not valid, the mark command is aborted and an error is returned to the syntax module (block 106). If the opcode is valid, the referenced mark is loaded into mark memory at the location of the mark pointer (block 108). If the opcode information extracted is a valid non-opcode (operand), then non-opcode mark information is loaded into mark memory at the location of the mark pointer (block 108). The pointers are incremented by the appropriate amount as provided in the lookup table, which is dependent on the byte size of the opcode and operand, if any (block 110). Then the pointers are compared to the address range to determine if the mark command is to be terminated (block 112). If the range is not completed, the state machine module returns to earlier step of extracting the opcode information from the read location specified by the current position of the read pointer (block 102), and continues in the same manner as listed above. If the range is completed, the state machine module indicates successful completion to the user and terminates the mark command and the state machine module, returning to the syntax module (block 114). *See Morrison, Col. 6, Lines 6-49.* Morrison, however, does not disclose removal of an identifier from a pointer, as admitted by the Office.

1 To correct this defect, the Office asserts *See*, and more particularly the
2 abstract of *See*, which is excerpted as follows:

3 A processor-implemented method is described for updating a
4 datum stored in a nonvolatile memory, bits of which cannot
5 be overwritten from a first logical state to a second logical
6 state without a prior erasure. A first storage location in the
7 memory that stores a first version of the datum is accessed. A
8 status field of the first storage location is checked to
9 determine whether the first version of the datum has been
10 superseded. If the status field of the first storage location
11 indicates that the first version of the datum has not been
12 superseded, then a most recent version of the datum is stored
13 in a second storage location of the memory. An address of the
14 second storage location is then written into a next location
15 address field of the first storage location and the status field
16 of the first storage location is written to indicate that the first
17 version of the datum has been superseded such that the datum
18 is updated without the prior erasure of the memory. If the
19 status field of the first storage location indicates that the first
20 version of the datum has been superseded, then the next
21 location address field of the first storage location is accessed
22 to obtain the address of a next storage location that stores a
23 second version of the datum that supersedes the first version
24 of the datum. The next storage location is then caused to be
25 the first storage location and the method repeats itself to
access the first storage location. An apparatus for storing a
datum is also described. *See See, Abstract.*

18 As is apparent from the Abstract of *See*, however, no such modification of a
19 pointer is disclosed.

20 **Claim 1** has been amended, and as amended (portions of the amendment
21 appear in bold/*italics* below), recites a method comprising:

- 22 • emulating an operation of a client; and
- 23 • permitting the emulated operation to access a contiguous portion of
24 emulated memory only when a pointer used by the emulated operation and
25 a table entry used to manage the emulated memory both contain the same
identifier, wherein an address to the contiguous portion is contained in both

1 the pointer and the table entry *and wherein the identifier is removed from*
2 *the corresponding pointer to permit the access to the contiguous portion*
3 *of emulated memory.*

4 Support for the amendment may be found throughout the specification and
5 drawings as filed, examples of which may be found in claims 3 and 29 as
6 originally filed. Neither Morrison nor See, alone or in combination, teach or
7 suggest these aspects. Withdrawal of the rejection is respectfully requested.

8 **Claims 2-5** depend either directly or indirectly from Claim 1 and are
9 allowable as depending from an allowable base claim. Each of the dependent
10 claims is allowable based on the same rationale discussed with respect to Claim 1.
11 These claims are also allowable for their own recited features which, in
12 combination with those recited in claim 1, are neither shown nor suggested in the
13 references of record, either singly or in combination with one another.

14 **Claim 6** has been amended, and as amended (portions of the amendment
15 appear in bold/italics below), recites a method comprising:

- 16
- 17 • making a call to a memory manager for an emulated memory access
18 operation to an allocated contiguous portion of emulated memory, wherein
19 a generation count has been assigned to:
 - 20 ▪ a plurality of table entries corresponding to a respective
21 plurality of said allocated contiguous portions of emulated
22 memory, and
 - 23 ▪ a plurality of pointers each containing an address to a
24 respective said allocated contiguous portion of emulated
25 memory;
 - comparing the generation count:
 - in the pointer containing the address to the allocated
contiguous portion of emulated memory; and
 - in the table entry corresponding to the allocated contiguous
portion of emulated memory;

- if the respective said generation counts in the comparison do not match, then outputting a diagnostic; **and**
- ***if the respective said generation counts in the comparison match, removing the generation count from the pointer specified by the memory manager for the emulated memory access operation during the performing of the emulated memory access operation for which the memory manager was called.***

Support for the amendment may be found throughout the specification and drawings as filed, examples of which may be found in claims 3 and 29 as originally filed. Neither Morrison nor See, alone or in combination, teach or suggest these aspects. Withdrawal of the rejection is respectfully requested.

Claims 7-8 and 10-13 depend either directly or indirectly from Claim 6 and are allowable as depending from an allowable base claim. Each of the dependent claims is allowable based on the same rationale discussed with respect to Claim 6. These claims are also allowable for their own recited features which, in combination with those recited in claim 6, are neither shown nor suggested in the references of record, either singly or in combination with one another.

Claim 14 has been amended, and as amended (portions of the amendment appear in bold/italics below), recites in a first computing device executing a first application for the emulation of a second computing device executing a second application, a method comprising:

- making a call from the second application to a memory manager for an emulated memory access operation to an allocated contiguous portion of emulated memory used by the second application and including a plurality of said allocated contiguous portions, wherein:
 - a generation count is in a plurality of table entries corresponding to a respective plurality of said allocated contiguous portions of emulated memory;

- a generation count is in a plurality of pointers each containing an address to a respective said allocated contiguous portion of emulated memory;
- for the emulated memory access operation, the memory manager uses the address in the pointer that corresponds to the allocated contiguous portion in emulated memory *after removal of the generation count from the pointer*; and
- prior to performing the emulated memory access operation to the allocated contiguous portion of emulated memory:
 - comparing the generation count:
 - in the pointer containing the address of the allocated contiguous portion of the emulated memory; and
 - in the table entry corresponding to the allocated contiguous portion of the emulated memory;
 - outputting a diagnostic when the respective said generation counts of the comparison do not match.

Support for the amendment may be found throughout the specification and drawings as filed, examples of which may be found in claims 3 and 29 as originally filed. Neither Morrison nor See, alone or in combination, teach or suggest these aspects. Withdrawal of the rejection is respectfully requested.

Claims 15-21 depend either directly or indirectly from Claim 14 and are allowable as depending from an allowable base claim. Each of the dependent claims is allowable based on the same rationale discussed with respect to Claim 14. These claims are also allowable for their own recited features which, in combination with those recited in claim 14, are neither shown nor suggested in the references of record, either singly or in combination with one another.

Claim 22 recites computer-readable medium containing instructions for execution by a computer, wherein the instructions comprise:

- first logic calling for an emulated memory access operation with respect to a first of a contiguous portion of an emulated memory for which there is:

- a corresponding table entry in a table having a plurality of said table entries that map to respective other said portions of the emulated memory, wherein each said table entry contains an identifier; and
- a corresponding pointer to a plurality of pointers each containing an identifier and an address to a respective said contiguous portion of the emulated memory;
- second logic, in response to the first logic, such that, if the identifier in the table entry corresponding to the first said contiguous portion is the same as the identifier in the pointer corresponding to the first said portion, then:
 - the emulated memory access operation is performed with respect to the first said contiguous portion of the emulated memory; and
 - when the emulated memory access operation is neither a read operation nor a write operation, the identifier is identically changed in both:
 - the table entry corresponding to the first said portion; and
 - the pointer corresponding to the first said portion;
- third logic, when the identifier in the table entry corresponding to the first said contiguous portion is different from the identifier in the pointer corresponding to the first said portion, calling for a diagnostic to be output.

Neither Morrison nor See, alone or in combination, teach or suggest these aspects.

For example, teaching or suggestion cannot be found in Morrison nor See, alone or in combination, for “when the emulated memory access operation is neither a read operation nor a write operation, the identifier is identically changed in both: the table entry corresponding to the first said portion; and the pointer corresponding to the first said portion”. Withdrawal of the rejection is respectfully requested.

Claims 23-25 depend either directly or indirectly from Claim 22 and are allowable as depending from an allowable base claim. Each of the dependent claims is allowable based on the same rationale discussed with respect to Claim

22. These claims are also allowable for their own recited features which, in combination with those recited in claim 22, are neither shown nor suggested in the references of record, either singly or in combination with one another.

Claim 26 has been amended, and as amended (portions of the amendment appear in bold/italics below), recites a first software program which, when executed by a computing device, emulates the execution of a second software program using emulated memory, the first software program comprising instructions that permit the second software program to perform an emulated memory access operation on a previously allocated contiguous portion of the emulated memory only when a pointer and a table entry both contain the same identifier, wherein:

- the pointer also contains an address to the previously allocated contiguous portion *which is useable to access the previously allocated contiguous portion after removal of the identifier*; and
- the table entry maps to the previously allocated contiguous portion.

Support for the amendment may be found throughout the specification and drawings as filed, examples of which may be found in claims 3 and 29 as originally filed. Neither Morrison nor See, alone or in combination, teach or suggest these aspects. Withdrawal of the rejection is respectfully requested.

Claims 27-31 depend either directly or indirectly from Claim 26 and are allowable as depending from an allowable base claim. Each of the dependent claims is allowable based on the same rationale discussed with respect to Claim 26. These claims are also allowable for their own recited features which, in

1 combination with those recited in claim 26, are neither shown nor suggested in the
2 references of record, either singly or in combination with one another.

3 **Claim 32** has been amended, and as amended (portions of the amendment
4 appear in bold/italics below), recites a computer-readable medium containing
5 instructions for execution by a computer, wherein the instructions comprise:

- 6 • means for emulating an operation of a client as the client executes an
7 application; and
- 8 • means for outputting a diagnostic when:
 - 9 ■ the emulated operation attempts to access a previously
10 allocated contiguous portion of emulated memory using a
11 pointer containing an identifier, *wherein the pointer is
12 configured to access the previously allocated contiguous
13 portion of the emulated memory upon removal of the
14 identifier*; and
 - 15 ■ a table entry used to manage the emulated memory does not
16 contain the same identifier as the identifier in the pointer,
17 wherein an address to the previously allocated contiguous
18 portion is contained in both the pointer and the table entry.

19 Support for the amendment may be found throughout the specification and
20 drawings as filed, examples of which may be found in claims 3 and 29 as
21 originally filed. Neither Morrison nor See, alone or in combination, teach or
22 suggest these aspects. Withdrawal of the rejection is respectfully requested.

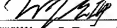
23 **Claims 33-36** depend either directly or indirectly from Claim 32 and are
24 allowable as depending from an allowable base claim. Each of the dependent
25 claims is allowable based on the same rationale discussed with respect to Claim
32. These claims are also allowable for their own recited features which, in
combination with those recited in claim 32, are neither shown nor suggested in the
references of record, either singly or in combination with one another.

1
2 **Conclusion**

3 All of the claims are in condition for allowance. Accordingly, Applicant
4 requests a Notice of Allowability be issued forthwith. If the Office's next
5 anticipated action is to be anything other than issuance of a Notice of Allowability,
6 Applicant respectfully requests a telephone call for the purpose of scheduling an
7 interview.
8

9 Respectfully Submitted,
10

11
12 Dated: 11/10/16

13 By: 
14 William J. Breen III
15 Reg. No. 45,313
16 (509) 324-9256 x249
17
18
19
20
21
22
23
24
25